Representational Requirements for a Plan Based Approach to Automated Camera Control

Arnav Jhala and R. Michael Young

Department of Computer Science
North Carolina State University
890 Oval Dr, Raleigh, NC - 27695
ahjhala@unity.ncsu.edu, young@csc.ncsu.edu

Abstract

Automated camera control has been an active area of research for a number of years. The problem has been addressed in the Graphics, AI and Game communities from different perspectives. The main focus of the research in the Graphics community has been frame composition and coherence. The AI community has focused on intelligent shot selection, and the Games community strives for real-time cinematic camera control. While the proposed solutions in each of these fields are promising, there has not been much effort spent on listing out the requirements of an intelligent camera control system and how these can be satisfied through a combination of approaches taken from these different fields. This paper attempts to list out the representational requirements with a view of finding a unifying representation for combining these disparate approaches. We show how a plan based approach can capture some of these requirements and it can be connected to a geometric constraint solver for camera placement.

Introduction

A virtual camera in a 3D world is a powerful communicative tool that conveys information about the virtual world. The camera can be used for exploration of the virtual world and for conveying the actions and event occurring within the world. Many recent games and training simulations have become narrative based. Stories told by such environments are dynamic in nature due to the interactivity of the medium. Giving the user of the environment complete control of the camera restricts the designer from effectively conveying the story. Games typically use pre-scripted cut-scenes at various time points in the user's exploration. In fully dynamic storylines like the ones generated by automated story generation systems like Mimesis (Young et. al. 2004), and Haunt (Magerko et. al. 2004), it is not possible to pre-script cut sequences.

For dynamic worlds, an automated camera planner needs to identify the salient elements of the world. It should then determine the possible shots (or sequences of shots) that could be used to film the salient elements.

In the medium of cinema, various stereotypical conventions have evolved from the experience of directors and cinematographers over the years. Such cinematic conventions or idioms have been documented by a number of film theorists and practitioners (Arijon 1976, Mascelli 1970, Monaco 1981). It has also been established by researchers that film techniques can be formalized into computational models like state machines (Christianson et. al. 1996) or as plan operators (Jhala et. al. 2005).

In this paper we will describe a formalism of concepts from film theory in a planning framework for an automated camera planning system. We use an existing discourse planner Crossbow, which is based on the Decompositional Partial Order Causal Link Planning algorithm presented by Young et. al (1994). We enrich Crossbow's representation with temporal assertions describing the actions unfolding in the story-world and the camera actions that film these actions in the 3D environment.

Actions, Objects and Time in Cinematography

To illustrate the relationship between actions happening in a 3D environment and camera actions, and to motivate the representational requirements of an automated system for generating such camera actions, consider the following part of the climax sequence from the movie *Rope* (Hitchcock, 1948) by Alfred Hitchcock. The scene contains 3 characters in a tense conversation. This sequence consists of a series of speech and movement actions of characters. The annotation to the script is show in Table 1. We will use the name C_i for camera actions and S_i, M_i and R_i for speak, move and react actions respectively.

M1. Rupert walks away from the table, shocked.
S1. Rupert speaks as he walks towards Brandon
R1. Brandon reacts with a surprised look.
M2. Rupert walks past Brandon
S2. Rupert speaks as he walks past Brandon

Rope solution (~ 72:00)									
	Scene			Character: Brandon			Character: Rupert		
Time	Mood	Intensity	Tempo	Emotion	Intensity	Action	Emotion	Intensity	Action
~ 72:00	Tense	Medium	Medium	Nervousness	Medium	Speak	Fear	Medium	Walk
				& Fear	Medium			Medium	React

Table 1: Table showing annotations to the script of character actions.

R3. Brandon is nervous and fearful in reaction to Rupert's speech.
As shown in Fig1, we can see that the camera action C1 maintains a three-quarter close-shot of Rupert as he speaks until the start of M1. Action C2's start time matches with the start of M1 when the camera zooms out and tracks

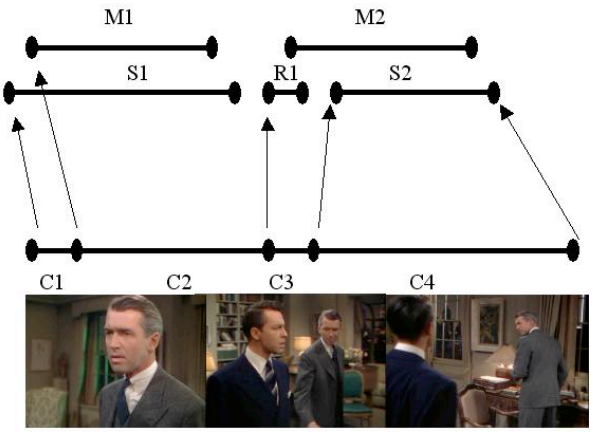

Figure 1: Relationship between camera actions and story world actions. The arrows indicate the start times and end-times of camera actions.

Rupert as he walks away from the desk. The following schedule was used to film the scene in the movie. We use the vocabulary used in (Arijon 1974) for describing the shots in this paper.

Schedule 1
C1. _ Close-Shot of Rupert who is shocked.
C2. Zoom out to Medium-Shot and track relative Rupert as he walks away from the desk.
C3. External Reverse shot of Rupert as he speaks to Brandon.
C4. Close 2-shot to film Brandon's reaction.

There are many different ways of filming the same sequence of actions by using a variety of shot types. An alternate schedule for the camera actions for filming the same conversation could be the following.

Schedule 2
C1. Long-2shot of both characters starting from the start of action S1to the end of S2.

The selection of a strategy for a particular sequence depends on the contextual parameters of the story, as well as the physical setting in which the story is unfolding. For example, in *Schedule 1* the selection of action *C1* is motivated by the intensity of emotion of the character. Maintaining low tempo can be achieved by selecting actions with fewer cuts or longer durations. Thus a zoom action (with longer duration) is chosen over a direct cut to a Medium-Shot in *C2* from a Close-Shot in *C1*. The decision of selecting the correct schedule depends on the rhetorical elements in the story. For instance, *Schedule 1* is typically chosen over *Schedule 2* when the interpersonal relationship of the characters is not intimate or the speech acts are not causally important to the other parts of the story (Mascelli 1970).

Stories in interactive games and simulations make the world more dynamic. The camera controller should be able to frame the shots in a cinematically acceptable way by taking into account the current physical state of the virtual world. For instance, in *C1* (Figure 1) a cinematic shot is generated by framing the character on three-quarters right of the screen. Also in *C3* and *C4* the movement of the character could be unpredictable in a dynamic environment so the camera should maintain the composition of the shot to account for such unpredictable motion.

Requirements for an automated camera planner

Based on the discussion in the previous section we can now start listing out the representational requirements for an automated camera placement system.

Req1: Story Representation: The foremost requirement of a camera planning system is a representation of the story unfolding in the virtual world that is being told to a viewer. Story world actions represented in just a sequence can lead to individual camera shots selected for filming actions, but they do not motivate selection of coherent and appropriate sequences of camera shots. Hence, the story representation should include the *causally related actions* happening in the story. The camera controller also requires the *durations and temporal relationships* between story world actions.
Req2: Physical World Representation: A camera in a dynamic 3D world needs to know the state of the physical world in order to account for occlusions and for framing of shots.

Req3: Rhetorical coherence: Discourse in any medium needs to follow a rhetorical structure in order to make the discourse actions coherent to the intended audience. A sequence of shots conveys coherent information if it is structured to satisfy certain rhetorical goals like Elaboration or Sequence. Camera placement should also take into account phenomena like suspense to include/exclude certain objects or events for showing the viewer. Certain actions or events in a story are more important than others, selection of discourse actions also depends on the relative importance of the actions to the story.

Req4: Temporal consistency: Camera shots must relate temporally to each other. This aspect is very important as duration of camera shots, and the relative ordering also affect the viewers. In film certain rules like *"180^0 rule"* and "*don't pan backwards"* need to be followed for camera actions executing in sequence temporally. Any camera placement system should keep track of the temporal consistency of the generated camera shots.

With these requirements in mind, we now describe how a planning formalism captures these requirements.

Story Representation

The story to be told in the virtual world is represented as a sequence of parameterized actions. We use the story plan representation described in (Young et. al. 2004) with the addition of temporal variables to the steps. The story consists of *steps*, *causal links*, and *ordering links* on the steps. Causal links specify the causal dependence of one step over another through an enabling condition. Ordering links represent the strict order in which certain actions execute in the world. Parameters of the steps specify the actors and objects or locations involved during the execution of these actions in the world. The discourse actions relate to the actions in the story world through temporal constraints on the execution times of the story world actions. Causal Links motivate certain rhetorical strategies for filming actions. Thus, such a representation satisfies *Req1* from the requirements list mentioned in the previous section.

Camera Actions in a Plan Representation

In order to fulfill the requirements for automated camera placement, discussed earlier, we formalize camera shots and shot sequences as hierarchical plan operators. We show how this formalism satisfies some of the requirements. These camera operators represent discourse strategies at different levels. At the highest level the operators represent abstract *Narrative Discourse Strategies* (like Show-Conflict and Show-Resolution). At a lower level these strategies (represented as abstract plan operators) get refined into more context sensitive *Scene-Level Actions* (like Show-Conversation). These action-level shot types are further refined into sequences of *Action-Level Camera Shots* (like LookAt-MediumShot, Internal-Reverse-Shot).

We have extended the representation of an existing planner Crossbow (Young et. al. 1994) to incorporate temporal indexes based on the representation discussed in (Nau et. al. 2005). In this representation each predicate or condition is annotated with a time interval represented by a start time (tstart) and an end time (tend).

Conditions: `(Predicate param*)@[tstart, tend)`

Operators: `(Action param*)@[tstart, tend)`

Relations: The planner supports temporal interval relations defined in (Allen 1980). These relations are used to specify constraints on the time variables in the conditions and operators. Following is the list of 7 relations.

(After ?a1 ?a2) = $tend_{a1} < tstart_{a2}$

(Before ?a1 ?a2) = $tstart_{a1} > tend_{a2}$

(during ?a1 ?a2) = $tstart_{a1} < tstart_{a2} < tend_{a2} < tend_{a1}$

(overlaps ?a1 ?a2) = $tstart_{a1} < tstart_{a2} < tend_{a1} < tend_{a2}$

(meets ?a1 ?a2) = $(tstart_{a1} = tstart_{a2}) \wedge (tend_{a2} = tend_{a1})$

(starts ?a1 ?a2) = $(tstart_{a1} = tstart_{a2})$

(finishes ?a1 ?a2) = $(tend_{a2} = tend_{a1})$

Handling Pragmatic Constraints with abstract operators

Every narrative is created with some high level goals of the author. In film too there are some goals that the director maintains throughout the unfolding of the plot. These high level goals are pragmatic in nature and are not satisfied by just the execution of a single action selected by a planner. These goals are similar to the goals such as *formality* in text generation systems such as Pauline (Hovy 1993). These goals require certain types of discourse actions to be maintained over certain periods of time. Such goals act as pragmatic constraints on the selection of operators during the planning process. Examples of such constraints are, the *Tempo* and *Emotion of characters* in the story. Here we show how a plan operator representation captures the process of maintaining pragmatic goals by encoding the cinematic idioms as decompositions on an abstract operator.

STRATEGY 1: Maintaining Tempo

High tempo is characterized by an increased number of transitions (cuts), and motion shots (pans, tilts). Lower tempo results in selection of shots with longer duration. In the example operator shown below, a low tempo value will lead to the refinement of the abstract conversation operator into a decomposition that contains only a 2-shot lasting the whole duration of the conversation. The start and end time

variables of the are temporally constrained to match with the times of the constituent actions in the conversation.

```
Operator: Show-Conversation @ [tstart, tend)
Parameters: ?c1 ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (conv-step ?c1 ?a1) (conv-step ?c1
?a2)
Temporal Constraints: (= ?tstart ?ta1start) …
Effects: (BEL V (Occurred ?c1)@[?tstart, ?tend))
IsAbstract: True

Decomposition: Show-Conversation
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (tempo low)@[?tstart, ?tend)
Temporal Constraints: (after step1 step2)
Steps: (step1 Master-Shot ?c1)
       (step2 Apex-Shot ?c1)
Effects: (BEL V (Occurred ?conv)@[?tstart,
?tend))
```

STRATEGY 2: Depicting Emotion

One of the ways of depicting heightened emotion is through high-angle (negative valence) or low-angle (positive valence) shots.

```
Operator: Show-Speaking @ [tstart, tend)
Parameters: ?s, ?tstart, ?tend
Preconditions:
Constraints: (speak ?s) (actor ?a ?s) …
Temporal Constraints: (= ?tstart ?ts1start) …
Effects: (BEL V (Occurred ?s1)@[?tstart, ?tend))
IsAbstract: True

Decomposition: Show-Speaking @[tstart, tend)
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (emotion high)@[t1, t2) (emotion
low@[t2, t3)
Temporal Constraints: (after step1 step2) (< t1
t2) (< t2 t3)
Steps: (step1 (Look-At-Close-Low ?a1 t1, t2))
       (step2 (Zoom-Out-Low-Med ?a1, t2, t3))
Effects: (BEL V (Occurred ?s1)@[?tstart, ?tend))
```

Automated text based discourse generation programs (Moore & Paris) have used this abstract operator formalism to encode rhetorical relationships between text segments. Similar rhetorical relationships also exist between camera shots. Since the representation that we use is similar to that used in traditional text based systems, the same rhetorical operators can be utilized for maintaining rhetorical coherence. Thus using hierarchical operators satisfies **Req3.**

Scene-Level Actions

Scene level actions are context sensitive actions that are selected as part of the refinement process on the abstract discourse actions. Scene level actions represent the boundaries of a single scene and can be further refined into a number of shots based on the constituent actions in the scene. Scene level operators also encode certain cinematic idioms. For instance the operators shown below encode the *Schedule 2* presented in one of the previous sections.

```
Operator: Show-Conversation
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints:
Effects: (BEL V (Occurred ?conv)@[?tstart,
?tend))

Decomposition: Show-Conversation
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (tempo low)@[?tstart, ?tend)
Temporal Constraints: (after step1 step2)
Steps: (step1 Master-Shot ?c1)
       (step2 Apex-Shot ?c1)
```

Action-Level Shots

Action level shots are the primitive shots that execute in the game engine. The operators for these shots mirror the shot types commonly used in cinema.

```
Operator: LookAt-Long-Low
Parameters: ?obj,  ?tstart, ?tend
Preconditions:
Constraints: !(focus ?obj Shot-Long
High)@[?tstart)
Effects: (focus ?obj Shot-Long Low)@[?tstart,
?tend)

Operator: Track-Actor-Absolute
Parameters: ?obj, ?shot-type, ?shot-angle,
?tstart, ?tend, ?tsetup, ?teasein
Preconditions: (focus ?obj ?shot-type ?shot-
angle)@[?tstart)
Constraints: (> 10 (- ?tend ?tstart))
Effects: (tracked ?obj ?shot-type ?shot-
angle)@[?tstart, ?tend)
```

Cam-Plan (P_C = <S, B, O, L_C, L_D, L_A>, Λ, Δ)
Here P_C is a partial plan. Initially the procedure is called with S containing placeholder steps representing the initial state and goal state and O containing a single ordering constraint between them requiring the initial state step to precede the goal state step. S also contains the steps that occur in the story, generated by a story planner.

Termination: If P_C is inconsistent, fail. Otherwise if P_C is complete and has no flaws then return P_C

Plan Refinement: Non-deterministically do one of the following

1. **Causal planning**
   a. **Goal Selection**: Pick some open condition p from the set of communicative goals
   b. **Operator Selection**: Let O be the operator with an effect e that unifies with p. Add a new step S_{add} as an instance of O and update the causal and temporal links <S_{add}, e, p, S>. S=S∪S_{add}, L_a=L_a∪<S_{wi}, S_{wj}, T, $S_{add,}$ e)
2. **Episode Decomposition**
   a. **Action Selection**: Non-deterministically select an unexpanded episode from P_C
   b. **Decomposition Selection**: Select an idiom for the chosen episode and add to P_C the steps and object as well as temporal constraints specified by the operator as the sub-plan for the chosen episode.

Conflict Resolution

A step S_t *threatens* the causal link L =<S_i, S_j, C> if S_t might possibly occur between S_i and S_j and S_t asserts a condition as an effect that unifies with C. For each threat in P_C created by the causal or episodic planning above, resolve the threat by nondeterministically choosing one of the following procedures:

Promotion: Move S_j before S_t if the ordering constraints are not violated

Demotion: if S_t before S_j if the ordering constraints are not violated

Variable Separation: add variable binding constraints to prevent the relevant conditions from unifying

Temporal Consistency Checking

The temporal constraints of newly added step are checked for consistency with the current temporal constraint list. If the constraints are consistent then they are added to the list.

Recursive Invocation

Call Cam-Plan with new value of Pc

Figure 2: Camera Planning Algorithm

DPOCL-T: Discourse Planning with Temporal Assertions

A Decompositional Partial Order Causal Link Planner with Temporal extensions can satisfy some of the representational requirements of a camera planning algorithm. The algorithm (Figure 2) is based on a modified version of an existing DPOCL planner Crossbow (Young et. al. 1994). Crossbow support to handle temporal predicates as described in the previous sections has been added. Given an input plan describing the actions in a story, and discourse goals, our system uses the algorithm shown in Figure 2. The algorithm is described in detail in (Young et. al. 1994).

Causal Planning takes place in the planner as described in (Young et. al. 1994), but with temporal variables. A precondition *p@[t)* is required to be true at the start of the action (t=tstart). Thus if effects of an action A1 satisfies the precondition of another action A2, then a temporal constraint of type is added for $tend_{A1} < tstart_{A2}$ such that no $p@[tend_{A1,} tstart_{A2})$.

Temporal Links are added between time points of story world actions and camera actions. If a camera action C@ [tstart, tend) films story world actions S1 and S2 then tstart is codesignated with $tstart_{S1}$ and tend is codesignated with $tend_{S2}$.Temporal Consistency is checked for each resolving step by ensuring that the start and end time of the added step is consistent with the temporal constraints in the constraint list.

An evaluation of this algorithm is beyond the scope of this paper. The representation and algorithm, however, are based on techniques that have been successfully implemented in the field of discourse generation in AI.

Managing Execution

For managing the execution of story and camera actions, execution manager receives a narrative plan with the story world actions as well as discourse actions. The execution manager starts executing the story-world actions as well as the camera actions that are linked relative to the start and end times of story world actions, at the appropriate times. We use the architecture similar to that described in (Young et.al. 2004) for an implementation of this algorithm. The operators mentioned in the previous sections can be executed as action classes within the game engine. Existing geometric constraint solvers (Bares et. al. 1999) can be connected to the output of the camera planning algorithm to satisfy the **Req. 3** for real-time frame composition.

Related Work

Camera Control in Virtual Environments

Computer Graphics researchers (Drucker 1994, Bares et al. 1999) have addressed the issue of automating camera placement from the point of view of geometric composition. The virtual cinematographer system developed by (Christianson et al. 1996) models the shots in a film idiom as a finite state machine that selects state transitions based on the run-time state of the world. The idioms are defined using a Declarative Camera Control Language (DCCL). (Tomlinson et al. 2003) have used expressive characters for driving the cinematography module for selection of shots and lighting in virtual environment populated with autonomous agents. More

recently, the use of neural network and genetic algorithm based approaches to find best geometric compositions (Hornung 2003, Halper 2004) have been investigated. (Kennedy 2002) uses rhetorical relations within an RST planner to generate coherent sequences. This approach does not exploit the causal relationships in plan actions and the reasoning is carried out locally for mood at every action.

Previous approaches to camera control in virtual environments have been restricted to finding the best framing for and determining geometrically smooth transitions of shots. Even the approaches that do exploit film idioms (Christianson et. al. 1996) only look at idioms related to composition. They have not attempted to exploit the narrative structure and causal relationships between shot or scene segments that affect the selection of camera positions.

Planning Coherent Discourse

Research in generation of coherent discourse in the field of artificial intelligence has focused on the creation of text. Planning approaches (e.g., (Young et al.1994, Maybury 1992, Moore & Paris 1989)) have been commonly used to determine the content and organization of multi-sentential text based on models developed by computational linguists (Grosz & Sidner 1986, Mann & Thompson 1987). Communicative acts that change the beliefs of a reader are formalized as plan operators that are chosen by a planner to achieve the intentional goals of the discourse being conveyed.

Acknowledgements

This research was supported by NSF CAREER Award #0092586.

References

Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *CACM*, 26(11):832-- 843, 1983.

Arijon, Daniel (1976). *Grammar of Film Language*, Los Angeles, Silman-James Press, 1976

Bares W. and Lester, J. (1997). Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments, *Sixth Conference on UM 1997*

Christianson David, Anderson Sean, He Li-wei, Salesin David, Weld Daniel, Cohen Michael (1996). Declarative Camera Control for Automatic Cinematography, *Proceedings of AAAI, 1996*

Drucker Steven, Zelter David (1997) Intelligent Camera Control in a Virtual Environment, *Graphics Interfaces 97*

Grosz, B. and Sidner, C. (1986). Attention, Intention and Structure of Discourse, *Proceedings of ACL* 1986

Halper, N., Helbing, R. and Strothotte, T. (2001). A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence. *Proceedings of Eurographics 2001*

Hitchcock, Alfred (1948). Rope, http://www.imdb.com/title/tt040746

Hornung, A., Lakemeyer, G., and Trogemann, G. (2003). An Autonomous Real-Time Camera Agent for Interactive Narratives and Games. *IVA 2003*

Hovy, E (1993). Automated Discourse Generation Using Discourse Structure Relations. *Artificial Intelligence* 63, pp. 341-385, 1993

Jhala, Arnav and Young, R. Michael. (2005). A Discourse Planning Approach for Cinematic Camera Control for Narratives in Virtual Environments. In *The Proceedings of the 25th AAAI*, Pittsburgh, PA, 2005.

Kevin Kennedy, Robert Mercer (2002). Planning animation cinematography and shot structure to communicate theme and mood, *Smart Graphics*, 2002.

Mascelli Joseph (1970) *The Five C's of Cinematography*, Cine/Grafic Publications, 1970.

Maybury M (1992). Communicative acts for explanation generation, *IJMMS* (1992) 37, 135-172

W. C. Mann, S. A. Thompson (1987). Rhetorical Structure Theory: A Theory of Text Organization. *TR- ISI/RS-87-190*, USC ISI, Marina Del Rey, CA., June 1987

Monaco James (1981). *How To Read A Film*, New York, Oxford University Press, 1981

Moore, J.D., Paris, C.L. (1989). Planning text for advisory dialogues. *In Proceedings of the 27th Annual Meeting of the ACL*, pg. 203--211, Vancouver, B.C., Canada, 1989.

Tomlinson Bill, Blumeberg Bruce, Nain Delphine (2000). Expressive Autonomous Cinematography for Interactive Virtual Environments *Fourth International Conference on Autonomous Agents*, Barcelona, Spain 2000.

Young R M., Moore, J. (1994). DPOCL: A Principled Approach To Discourse Planning, *Proceedings of the INLG workshop*, Kennebunkport, ME, 1994

Young, R. M., Riedl, M., Branly, M., Jhala, A., Martin, R.J. and Saretto, C.J. (2004). An architecture for integrating plan-based behavior generation with interactive game environments, *JOGD 1*, March 2004.